I claim:

1. A method of preventing stack manipulation attacks during function calls, which comprises, in an event of a call of an unsafe function defined in a given stack area, restricting stack access by hardware to the given stack area of the unsafe function.

2. The method according to claim 1, wherein the step of restricting stack access comprises storing a reference to a stack frame of a calling function before the call of the unsafe function.

3. The method according to claim 2, which comprises providing a mechanism preventing the called function from being able to access the value of the reference, the stack frame, and all data lying before that stack frame.

4. The method according to claim 1, which comprises providing a protective mechanism to ensure that the stack pointer does not go beyond the valid stack area of the called function.

5. The method according to claim 1, which comprises restoring the stack to an original state upon returning from an unsafe function.

6. The method according to claim 1, which comprises, in an event of a function call, initially reserving a memory area on the stack for function data to be protected, and thereafter optionally placing function arguments on the stack, and placing the reference, lying in the protected area, to the stack frame of the calling function on the previously reserved area of the stack, and writing the reference to the stack frame of the called function into the protected area.